

DICOM Web Viewer

BACHELOR'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science

in

Medical Informatics

by

Michael Kaserer

Registration Number 1025263

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: Ao.Univ.Prof. Univ.–Doz. Dipl.–Ing. Dr.techn. Eduard Gröller
Assistance: Dipl.–Ing. Gabriel Mistelbauer

Vienna, 8th July 2013

(Signature of Author)

(Signature of Advisor)

Erklärung zur Verfassung der Arbeit

Michael Kaserer
Haizingergasse 13/14, 1180 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 8. Juli 2013

(Ort, Datum)

(Unterschrift Verfasser)

Kurzfassung

Das Ziel dieser Arbeit ist die Implementierung eines Open Source web-basierten DICOM-Viewers. Web-Technologien werden bereits erfolgreich in der medizinischen Informatik verwendet. Da jedoch medizinische Software meist proprietär und auf bestimmte Betriebssysteme beschränkt ist, ist die Kombination von Web-Technologien zusammen mit DICOM ein logischer nächster Schritt. Aus diesem Grund wird mit dieser Arbeit versucht, die Grundlage für ein solches Open Source-Projekt zu legen.

Digital Imaging and Communication in Medicine (DICOM) ist ein offener und etablierter Standard zur Speicherung und zum Austausch von Informationen im medizinischen Bilddatenmanagement. DICOM definiert nicht nur ein Datenformat, sondern spezifiziert auch ein Kommunikationsprotokoll basierend auf dem ISO/OSI Modell zum Austausch von medizinischen Bildern. Diese Arbeit gibt zunächst einen Überblick über die Entstehungsgeschichte und beschreibt die wichtigsten Konzepte des Standards. Health Level 7 (HL7) ist ein Industriestandard für den Datenaustausch zwischen unterschiedlichen Krankenhausinformationssystemen. Integrating the Healthcare Enterprise (IHE) ist eine Initiative mit dem Ziel den Datenaustausch zwischen Informationssystemen im Gesundheitswesen mit verbreiteten Standards wie DICOM und HL7 zu harmonisieren. Anhand eines klinischen Workflows wird erklärt, wie DICOM und Health Level 7 mit Hilfe von IHE interagieren.

Der DICOM Viewer wurde als “Rich Internet Application” konzipiert und in JavaScript, HTML5 und CSS3 entwickelt. Um ein einfach bedienbares User Interface zu erstellen, wurden bereits bekannte DICOM Viewer analysiert und gemeinsame Merkmale übernommen. DICOM Dateien werden direkt im Browser mit JavaScript verarbeitet und mit Hilfe des HTML5 <canvas> Elementes visualisiert. Der Source Code wird im Anhang detailliert beschrieben und alle eingebundenen Bibliotheken aufgelistet.

Abschließend kann man sagen, dass Web-Technologien bestens zur Darstellung von DICOM-Bildern in einem Web-Browser geeignet sind. Besonders die plattformübergreifende Benutzung macht den DICOM Web Viewer vielversprechend für die Zukunft und könnte die traditionell eingesetzte Software ersetzen.

Abstract

The aim of this thesis is to introduce a web-based and open source DICOM viewer. Web technologies have already been successfully used in medical informatics. Since software used for medical applications is commonly proprietary and mostly restricted to certain operation systems, the combination of web technologies together with the DICOM standard is a natural next step. For this reason, this Bachelor's Thesis attempts to lay the basis for such an open source project.

Digital Imaging and Communication in Medicine (DICOM) is an established and non-proprietary standard for the storage and the exchange of information in medical imaging. DICOM defines not only a data format, but it specifies also a network protocol based on the ISO/OSI-model. This work provides first an overview of the DICOM's history and describes its main concepts. Health Level 7 (HL7) is an industry standard for the data exchange between different hospital information systems. Integrating the Healthcare Enterprise (IHE) is an initiative that aims to harmonize the exchange of data between information systems in health by promoting common standards such as DICOM and HL7. Based on a clinical workflow, the interaction of DICOM and HL7 by using IHE is shown.

The DICOM Web Viewer was designed as a "Rich Internet Application" and developed in JavaScript, HTML5 and CSS3. To create an easy to use user interface, state-of-the-art DICOM viewers were investigated and a common set of features were identified. DICOM files are parsed directly in the browser using JavaScript and then visualized using the HTML5 <canvas> element. In the appendix the source code is described in detail and all third-party libraries get listed.

The implementation shows, that web technologies are well suited to view DICOM images in a web browser. Especially its cross-platform usability makes the DICOM Web Viewer even more promising for the future and may replace traditional proprietary software.

Contents

1	Introduction	1
1.1	Motivation	1
2	DICOM Standard	2
2.1	Short History	2
2.2	Data Format	3
2.3	Service Class	6
2.4	HL7 and IHE	8
3	Imaging Workflow	10
3.1	HIS	11
3.2	RIS	11
3.3	Modality	11
3.4	PACS	11
3.5	Review Workstation	11
4	DICOM Web Viewer	12
4.1	Features	12
4.2	Architecture	13
4.3	Technology Stack	13
4.4	User Interface	14
5	Conclusion and Future Work	19
5.1	Conclusion	19
5.2	Future Work	19
A	Source Code	21
A.1	File Structure	21
A.2	License	23
A.3	Third-party Libraries	24
	Bibliography	25

List of Abbreviations

ACR	American College of Radiology
API	Application Programming Interface
CSS	Cascading Style Sheet
CT	Computed Tomography
DICOM	Digital Imaging and Communication in Medicine
DIMSE	DICOM Message Service Element
DOM	Document Object Model
EMR	Electronic Medical Record
GNU	GNU's Not Unix
GPL	General Public License
HIS	Hospital Information System
HL7	Health Level 7
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IE	Information Entity
IHE	Integrating the Healthcare Enterprise
IOD	Information Object Definition
IP	Internet Protocol
ISO	International Organization for Standardization
JS	JavaScript
MIT	Massachusetts Institute of Technology
NEMA	National Electrical Manufacturers Association

OSI	Open Systems Interconnection
PACS	Picture Archiving and Communication System
RIA	Rich Internet Application
RIS	Radiology Information System
SCP	Service Class Provider
SCU	Service Class User
SOP	Service–Object Pairs
TCP	Transmission Control Protocol
VL	Value Length
VM	Value Multiplicity
VR	Value Representation
W3C	World Wide Web Consortium
WADO	Web Access to DICOM Persistent Objects
WC	Window Center
WW	Window Width

Introduction

1.1 Motivation

DICOM is the abbreviation for Digital Imaging and Communications in Medicine and is a non-proprietary standard for handling, storing, printing and transmitting information in medical imaging. Since the inception of this standard some 30 years ago, it has become the universal standard of digital medicine and the driving force behind the entire medical imaging workflow. All current digital image-acquisition devices produce DICOM images and communicate through DICOM networks. DICOM does not only store the images, but it also records a multitude of other image-related parameters such as physical sizes of objects in the image, slice thickness or image exposure parameters. These data enrich the informational content and facilitate interpretation and post-processing of medical images.

Web technologies emerged rapidly during the last decade and enabled fast and reliable access to all kinds of information. Nowadays web applications got all prerequisites to compete with traditional desktop applications. The aspect of platform independence and accessibility from almost every device makes web applications even more promising for the future. Since software used for medical applications is commonly proprietary and mostly limited to a specific operating system, the combination of web technologies together with the DICOM standard is a natural next step. Although there are some DICOM web viewer implementations available, these are either only commercial products or ageing projects.

The aim of this Bachelor's Thesis is to lay the basis of an open source web-based DICOM viewer project, keeping a focus on cross-platform usability and the use of modern web technologies like HTML5, CSS3 and JavaScript. The integration of these technologies allow a platform independent use of the viewer, which requires no client installation and can be used from any web-capable device. The project should be open source and released under the GNU General Public Licence (GPL).

DICOM Standard

The subsequent sections provide a short overview of DICOM's history and describe its fundamental concepts. The last part of this chapter discusses the relation of DICOM to the industrial standard HL7 and to the IHE initiative.

Two fundamental components of DICOM are the Information Object Class and the Service Class. The Object Class defines the content of a set of images and the Service Class describes what to do with these objects. The Service Classes and Information Object Classes are combined to form the fundamental units of DICOM *Service-Object Pairs (SOP)* [5].

2.1 Short History

The development of the DICOM standard is based on the cooperation of the American College of Radiology (ACR) with the National Electrical Manufacturers Association (NEMA). In 1983 a joint committee was founded to create a possibility for data transfer regardless of manufacturers' standards and to facilitate the development and expansion of Picture Archiving and Communication Systems (PACS).

As a first result the ACR-NEMA standard version 1.0 was published in 1985 and enabled data archiving on media and communication in a non-proprietary form. The first version was succeeded by revisions in 1986 and 1988. The latter became public as ACR-NEMA version 2.0 and introduced a new command support for display devices, new hierarchy scheme to identify images, and more specific data elements. These early standards specified a hardware interface, a minimum set of software commands and a consistent set of data formats.

The third version of the standard was released in 1993. Its name was changed to "DICOM" to improve the acceptance as an international standard. The concept of Service Classes, the support of offline media environment and the Conformance Statement was introduced. The latter is a formal statement associated with a specific implementation and specifies which parts of the DICOM standard are supported by the implementation. The essential difference was the specification of a network protocol relying on the ISO/OSI-model and the use of TCP/IP to

enable independence of vendor-specific solutions. The data structure was based on a model with unique identifiers for services and objects [8, 10].

The third version of the standard is still the latest, but it has been constantly updated and extended since 1993. With the release of version 3.0 DICOM dropped the use of the version number and started to version-number the standard using the release year. In our application the 2011 version of DICOM was used. The DICOM standard has reached a near universal level of acceptance amongst medical imaging equipment vendors and healthcare IT organizations, but has also its limitations: It is a standard directed at addressing technical interoperability issues in medical imaging and not a framework for achieving a useful clinical workflow. The Integrating the Healthcare Enterprise (IHE) initiative addresses this issue and defines therefore a layer on top of common standards like DICOM and Health Level 7 (HL7) [15].

2.2 Data Format

This section explains two important topics of the DICOM Data Format: the Real World Information Model and the DICOM File Format. The former is used to define the hierarchical data structure from patient, to studies, series, and images. The latter describes how to encapsulate a DICOM file ready for a SOP service [5].

Real World Information Model

DICOM uses its own terminology to describe its Real World Information Model. All real-world data, like patients, medical devices, studies or series, are viewed by DICOM as objects with its own properties and attributes that are standardized according to DICOM *Information Object*

Patient in
Real World

Patient object in
DICOM World

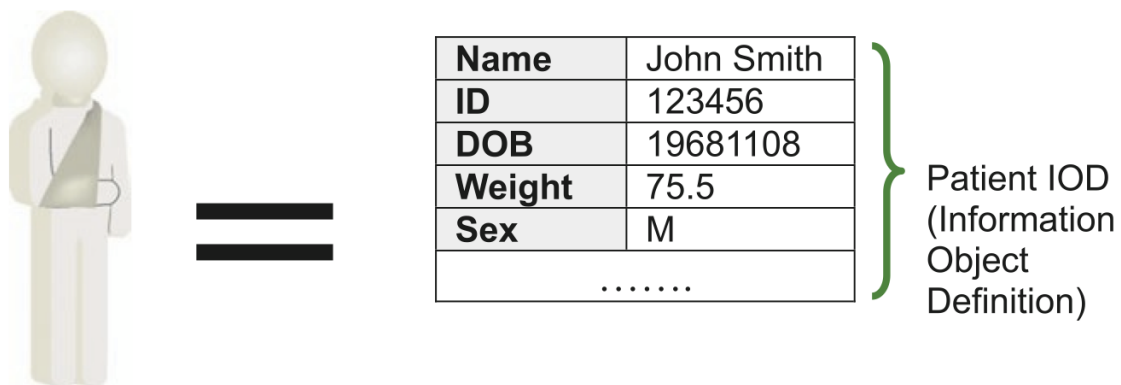


Figure 2.1: From real data to DICOM Information Object Definitions (IODs). Each IOD is a collection of attributes or real-world objects [12].

Definitions (IODs). An IOD is an object-oriented abstract data model used to specify information about real-world objects [11]. A patient IOD, for example, has attributes for patient's name, ID, date of birth, weight, sex and so on. All clinically relevant patient related information can be described (see Figure 2.1).

The DICOM model defines four object levels: patient; study; series & equipment; image, waveform and structured report document. Each of the levels can contain several sublevels. A patient is a person receiving health care services. One patient may have multiple studies and each study may include one or more image series with one or more images. Each level of the hierarchy has a unique ID to allow performing hierarchical data searches, retrievals and transactions (see Figure 2.2) [5].

DICOM differentiates between normalized and composite IODs. An normalized IOD represents a single real-world entity with attributes that are inherent in the real-world entity. For example, the DICOM Study IOD is normalized, because it contains only inherent properties from the study such as study date and time. Composite IODs are mixtures of several real-world entities. A CT image IOD, for example, contains attributes of the patient (name, ID, etc.) and

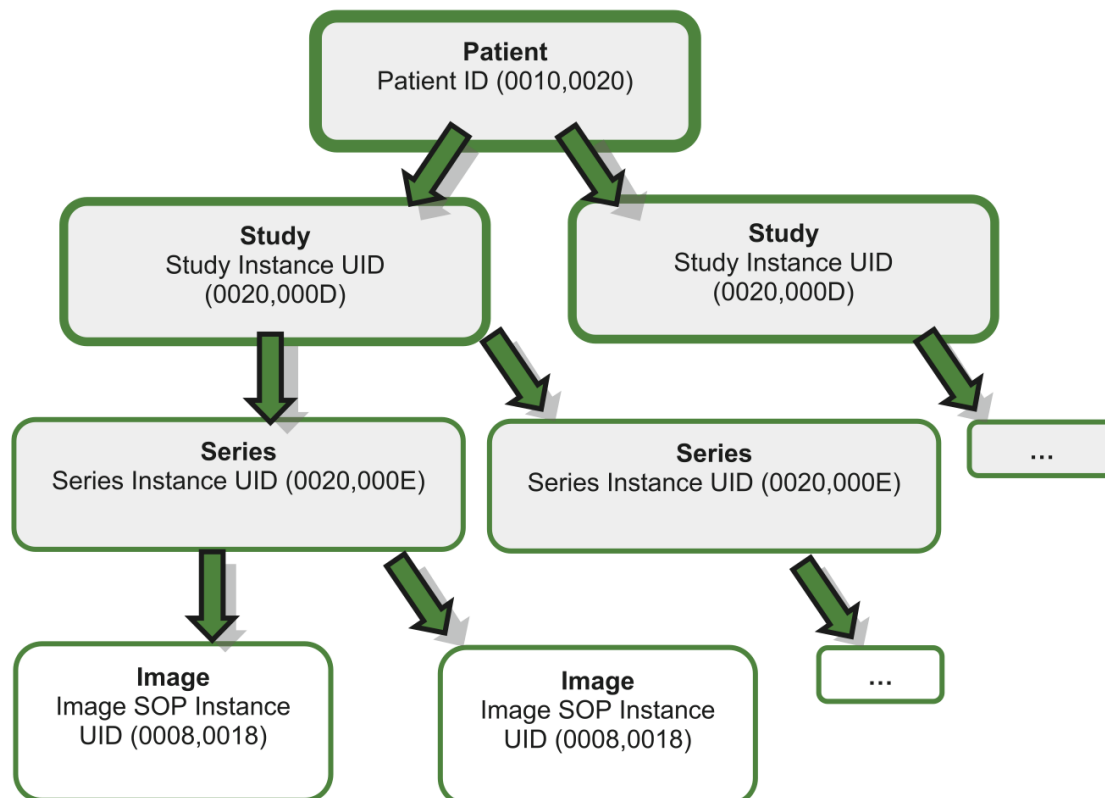


Figure 2.2: Four levels of DICOM information hierarchy. Each level is identified by its key ID. One patient may have multiple studies and each study may include one or more image series. Each series may contain one or more images [12].

attributes of the CT scanner and more. This combination of several real-world entities makes a CT image IOD composite [12].

The DICOM *Information Modules* are used to group attributes into logical and structured units. For example, the patient module contains all relevant information about the patient, such as name, ID, gender or age. Modules can be mandatory, conditional or user-defined and may be reused in different composite objects [16]. DICOM *Information Entities (IEs)* represent real-life entities and are built from DICOM Information Modules. The DICOM standard lists for each IE the modules it should include. The common patient IE, for example, should include the patient module, the specimen identification module and the clinical trial subject module.

The attributes of an IOD describe the properties of a real-world object instance and are encoded as *Data Elements* [12].

File Format

DICOM objects are encoded in binary form as lists of attributes. Each attribute can uniquely be identified by a tag consisting of a 16-bit group and 16-bit element number. The data elements are sorted sequentially by ascending tag number. Standard DICOM elements have an even group number and private data elements have an odd group number. The value field is padded to an even byte length and its meaning is defined by the IOD in which it is used. The *Value Representation (VR)* is defined by the Data Dictionary and describes whether it is a date, binary value, string, name or other. The Data Dictionary defines the VR for all the standard data elements as well as their *Value Multiplicity (VM)* and is not transmitted with the data set. The user of the DICOM standard has to know that the data element (0010, 0010) represents patient's name. The *Value Length (VL)* describes the length of the value in bytes. This is illustrated in Figure 2.3.

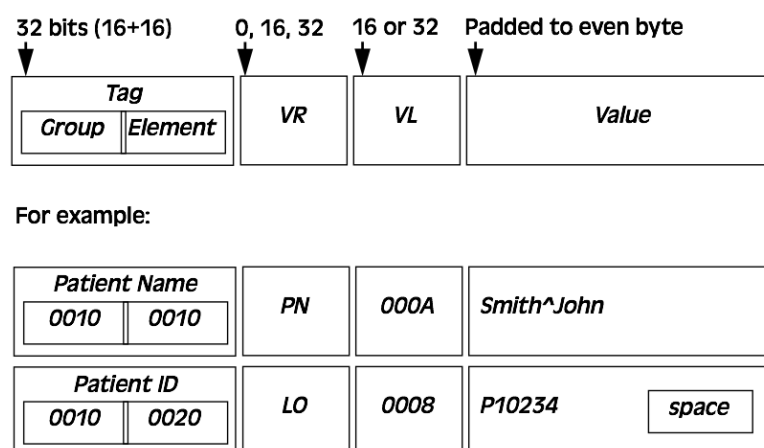


Figure 2.3: Data Element encoding. Each attribute is identified by a 16-bit group and 16-bit element number. Value Representation (VR) describes the type and Value Length (VL) the length in bytes of the value. The value field is padded to an even byte length [3].

In the application header it's defined whether *explicit* or *implicit* VRs are used. Explicit VRs specify its type together with each object, whereas the type of implicit VRs has to be looked up from a dictionary. This dictionary then maps tags (group, element) to VR values. The DICOM standard defines the implicit VR as the default encoding or transfer syntax. All DICOM implementations must support this default transfer syntax [3].

Images follow the same data-encoding rules. Each medical image has several important elements that correspond to attributes in the standard DICOM Data Dictionary. Some of them are:

- Image height: corresponds to the “Rows” attribute (0028, 0010)
- Image width: corresponds to the “Columns” attribute (0028, 0011)
- Image pixel data: corresponds to the “Pixel Data” attribute (7FE0, 0010)

The Pixel Data attribute contains the actual image and may takes up to 95% of the actual DICOM file size. Sequences of image frames can also be put into a single DICOM object, which is how DICOM stores digital video such as multiframe images or ultrasound cine loops [12].

2.3 Service Class

DICOM services are used for communication of imaging information objects within a device and between devices. A service is built on top of a set of *DICOM Message Service Elements (DIMSEs)*. There exist DIMSEs for normalized objects and DIMSEs for composite objects. The composite commands are generalized, whereas the normalized commands are more specific. DIMSEs are paired: when a device issues a command request, the receiver responds to the command accordingly.

Command	Function
N-GET	Retrieval of information object attribute value
N-SET	Specification of information object attribute value
N-CREATE	Creation of an information object
N-DELETE	Deletion of an information object
C-STORE	Transmission of an information object instance
C-FIND	Inquiries about information object instances
C-GET	Transmission of an information object instance via third-party application processes
C-MOVE	Similar to GET, but end receiver is usually not the command initiator

Table 2.1: Selected DICOM message service element (DIMSE). Commands with a “N” prefix are normalized and those with a “C” prefix are composite DICOM Message Service Elements (DIMSEs).

Because of the object-oriented nature of its information structure model, DICOM services are referred to as “Service Classes”. A *Service Class Provider (SCP)* provides a service, whereas a *Service Class User (SCU)* uses a service. For example, a harddisk in the PACS server is a service class provider for the server to store images. On the other hand, a CT scanner is the service class user of the harddisk in the PACS server to store images. A device can be either a service class provider or a service class user or both, depending on how it is used. Selected DICOM Services and their functions are outlined in Table 2.1.

Communication

DICOM’s network communication is based on the International Standards Organization Open Systems Interconnection (ISO/OSI) for information transmission. The ISO/OSI-model characterizes the functions of a communications system by partitioning it into abstraction layers. The model groups similar communication functions into one of the seven layers, starting from the physical layer to the highest application layer. A layer is served by the layer below it and serves the layer above it.

A process is called a service, when information objects are sent between layers in the same device. However a process is called a protocol, when objects are sent between two devices and several steps are invoked in two devices [5]. Figure 2.4 illustrates the data flow of the CT images from the scanner to the archive using DICOM. The numerals represent the steps as follows:

1. The CT scanner encodes all images into a DICOM object

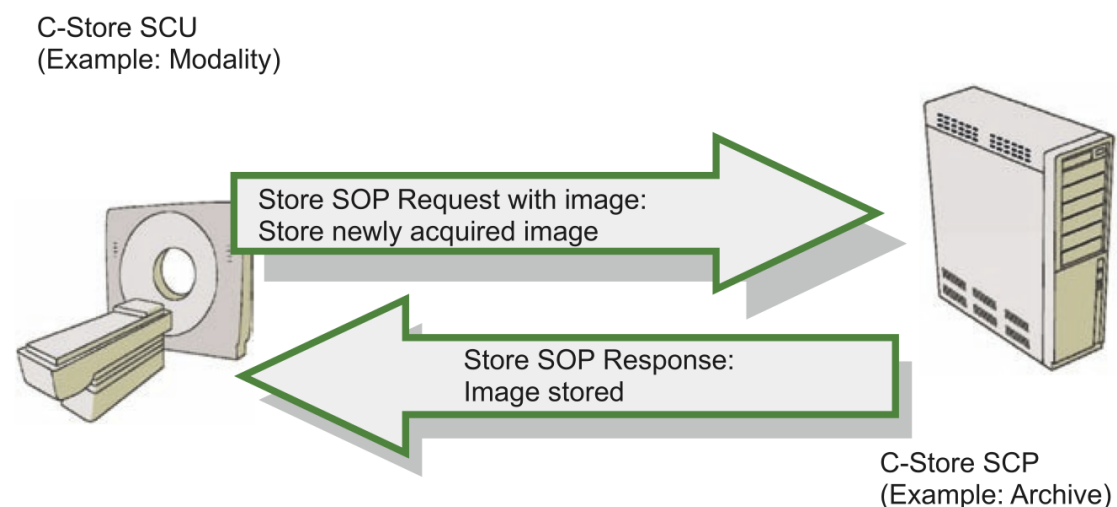


Figure 2.4: C-Store has a request part (sent from the C-Store SCU) and a response part (replied from the C-Store SCP). The request message will be immediately followed by the image to be stored. The response message returns then the status “stored successfully”, “still in the process of transmission” or eventual warnings/errors [12].

2. The scanner invokes a set of DIMSEs to move the image object from a certain level down to the physical layer in the ISO/OSI-model
3. TCP/IP is used for file transmission
4. The archive uses a counterset of DIMSEs to receive the image object through the physical layer
5. The archive stores the DICOM image and sends a response message back

2.4 HL7 and IHE

Transmission of information between healthcare information systems has always been difficult for two reasons. First, images and data are generated from various imaging modalities of different manufacturers and second, data is exchanged between information systems that use different computer platforms. *Integrating the Healthcare Enterprise (IHE)* is not a standard, but it drives the adoption of standards by promoting the use of DICOM and *Health Level 7 (HL7)*. With emerging industry standards like DICOM, HL7 and the IHE initiative on top, it has become feasible to integrate all these heterogeneous subsystems into a standardized system.

HL7

HL7 was established in 1987 and is a non-profit organization with the aim to develop a standard for electronic data exchange in healthcare environments and to simplify the interface implementation between computer applications from multiple vendors. The name “Health Level 7” is a reference to the seventh layer of the ISO/OSI-model, also known as the application layer, and indicates that HL7 focuses on application layer protocols for the healthcare domain, independent of lower layers. The standard defines data formats and protocols for exchanging textual data among healthcare information systems. Figure 2.5 shows how HL7 and DICOM are used to enable data exchange between different healthcare information systems during a generic radiology workflow.

HL7 develops conceptual standards (e.g. HL7 Reference Information Model), document standards (e.g. HL7 Clinical Document Architecture), application standards (e.g. HL7 Clinical Context Object Workgroup) and messaging standards (e.g. HL7 version 2.x and version 3.0) [5, 12, 17].

IHE

Even with the DICOM and HL7 standards available, there is still a need of a common consent on how to use these standards for integrating heterogeneous healthcare information systems. IHE is an initiative founded by healthcare professionals and industry in 1998 to improve the way computer systems in healthcare share information. IHE promotes the coordinated use of established standards such as DICOM and HL7 to address the specific clinical need in support of optimal patient care [6].

The IHE technical framework has three main key concepts: The *Data Model* describes the relationships between objects like patients, studies or orders. *IHE Actors* are not necessarily persons and exchange messages with other actors to achieve specific transactions and tasks. The latter are grouped into discrete units by *Integration Profiles*. Integration Profiles include the required IHE Actors and transactions to describe a particular clinical task or workflow [5]. An example is the scheduled imaging workflow described in chapter 3.

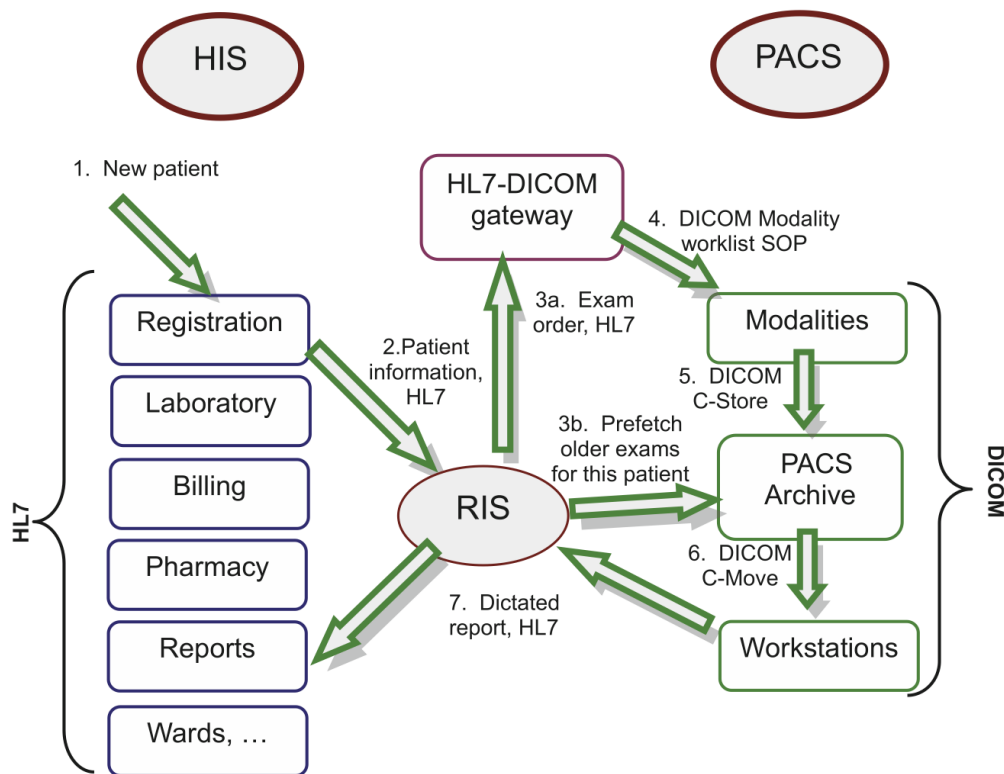


Figure 2.5: Interaction of DICOM and HL7 during a generic path of a scheduled radiology exam. Hospital Information System (HIS) and Radiology Information System (RIS) take care of processing anything but images. HIS works as the central repository for all patient-related information and RIS does the same job at the radiology department level. Both HIS and RIS use HL7 as data standard. The Picture Archiving and Communication System (PACS) stores and provides access to medical images and uses the DICOM standard [12].

Imaging Workflow

This section discusses a generic imaging workflow in medicine. Starting from the patient registering in the Hospital Information System (HIS), Radiology Information System (RIS) ordering examination, technologist performing the exam, image viewing, reporting, to image storing and archiving in the Picture Archiving and Communication System (PACS).

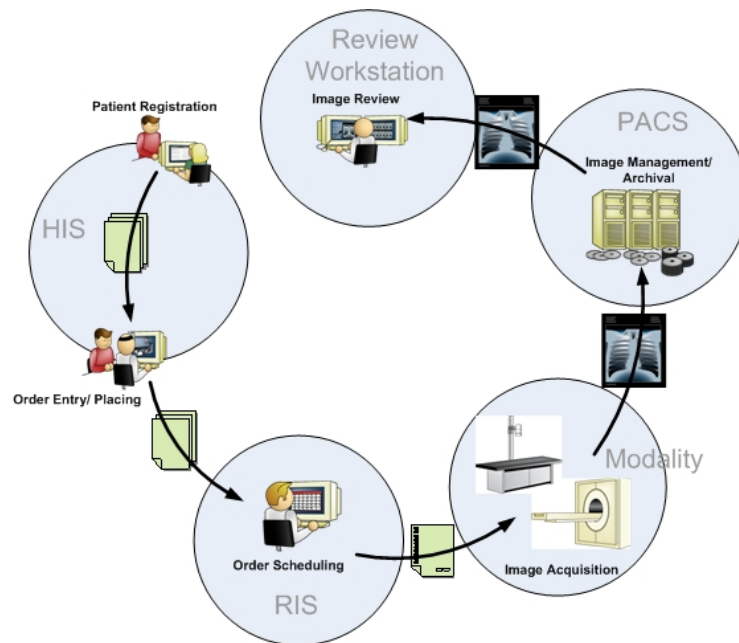


Figure 3.1: Imaging Workflow: Patient gets registered in the HIS and an order placement is sent to the RIS where it gets allocated to the specific modality. The acquired image is sent then to the PACS for storage and afterwards forwarded to the radiologist's workstation for diagnosis [7].

3.1 HIS

The workflow begins when the patient comes to the hospital and gets registered in the HIS. An Electronic Medical Record (EMR) is generated for the new patient. After being seen by a physician, additional information is added to the EMR and a medical procedure is ordered, e.g. an X-ray of the thorax. The order entry is then placed in the RIS.

3.2 RIS

The RIS generates a study scheduled event and sends a HL7-event to the HL7-DICOM gateway. This gateway translates the event from HL7 to DICOM and keeps the study scheduled event until a preset time, such as midnight the day before the study is scheduled. Then the gateway sends the study scheduled event to the PACS. Following on prefetching rules, previous exams of the scheduled patient are prefetched from the archive server and sent to the radiologist's workstation.

3.3 Modality

When the patient arrives at the modality, X-ray in our example, the technologist requests the DICOM worklist that contains the patient information and the details of the study to be performed. At the beginning and end of the study so-called "study started events" and "study completed events" are sent through the gateway component to the RIS.

3.4 PACS

After the images had been acquired, the technologist performs a quality check and forwards them to the PACS for storage. The PACS then verifies the images, stores and forwards them to the review workstation.

3.5 Review Workstation

The reporting phase occurs when the imaging case is complete and the radiologist views the study images and reports possible pathologies. At the workstation the radiologist searches in his worklist for studies to review. The selected study's images are then sent from the PACS to the workstation.

Radiologists usually use for reporting a voice dictation system. The recorded report is transcribed by a transcriptionist and verified by the radiologist. The report and the report status is sent to the RIS for storage. As the RIS receives the report, it sends update events to the HL7-DICOM gateway. These events are sent to the PACS to update the study status. The report is then added to the patient's EMR and physicians in the hospital can now query for the images and the report of the study [1, 12].

DICOM Web Viewer

This section provides a technical description of our developed DICOM Web Viewer. We will describe the specification, the architecture, the used technologies and some implementation aspects in details. Additionally we present screenshots of our application.

4.1 Features

In order to make our application easy to use for a broad audience, we investigated state-of-the-art DICOM viewers like DicomWorks¹, OsiriX², ResolutionMD³ or Siemens syngo fastView⁴ to identify a common set of features. Following requirements were identified:

- Load and display DICOM images by specifying a directory
- Sort images by patient's name and study's name
- Show sorted images in a tree-view
- Scrolling through the images of a series
- Changing the windowing function
- Zooming
- Moving
- Length measurement
- 1×2, 2×2 and 4×4 grid-view

¹<http://www.dicomworks.com/>

²<http://www.osirix-viewer.com/>

³<http://www.calgaryscientific.com/resolutionmd/>

⁴<http://healthcare.siemens.com/medical-imaging-it/syngo-special-topics/syngo-fastview>

- Show patient and study information in the corners
- Display all DICOM attributes of the image
- Reset functionality

A very important aspect was the cross-platform usability of the project: the DICOM Web Viewer should be accessible from any modern web browser (Firefox, Chrome, Safari, Opera, Internet Explorer, etc.) within different operating systems and without the need to install external plugins.

4.2 Architecture

The DICOM Web Viewer was designed as *client-side Rich Internet Application (RIA)*. A RIA combines the Web's lightweight distribution architecture with desktop applications' interface interactivity and computation power. It uses only built-in browser functionality to implement comparable interfaces [4].

Client-side refers to operations that are performed by the client. All operations are performed client-side because they require input by the user without sending data over the network, take less time, use less bandwidth and incur lesser security risks. The latter is a very important aspect that must be appropriately taken into consideration because DICOM files contain highly sensitive data.

4.3 Technology Stack

HTML5

HTML stands for “*HyperText Markup Language*” and is used to structure and present content on web pages. It is developed by the the World Wide Web Consortium (W3C) and the fifth revision of the standard has the status of “Candidate Recommendation” at the time of writing [13].

The new elements added to the version 5 of the HTML standard are key technologies for our viewer. Without the HTML5 <canvas> element, the new File API and the custom data attributes, it would not have possible to implement a web-based and plug-in-less viewer.

CSS3

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. It enables the separation of document content (e.g. written in HTML) from document presentation. The CSS specifications are also maintained by the W3C [14].

JavaScript

JavaScript (JS) is an interpreted scripting language that is dynamically, weakly typed, object-oriented and classless. It is implemented as part of web browser so that client-side scripts can

interact with the user, control the browser, communicate asynchronously and generate, alter or reload the document content, thus expanding the capabilities of HTML and CSS [18].

Our viewer uses JavaScript to parse and read DICOM files, perform a transformation from 16bit to 8bit pixel values and draw them on the HTML <canvas> element. Beyond that several third-party JavaScript libraries are used in the project (see appendix A for entire list).

4.4 User Interface

The viewer has two main parts: the sidebar on the left is used to select a directory that contains DICOM images and to display the images in a tree-view, whereas the right section displays the selected studies (Figure 4.1). Functionalities like move, zoom, change of windowing function and length measurement can be applied on them.

When the user selects a directory, all DICOM files in it are parsed and then displayed in the sidebar. The files are sorted by patient's name and study's name and grouped as nested lists. Those lists are then rendered in a expandable and collapsible tree by the jqueryTree library. The studies of a certain patient can be opened by or closed by clicking on its parent list item -

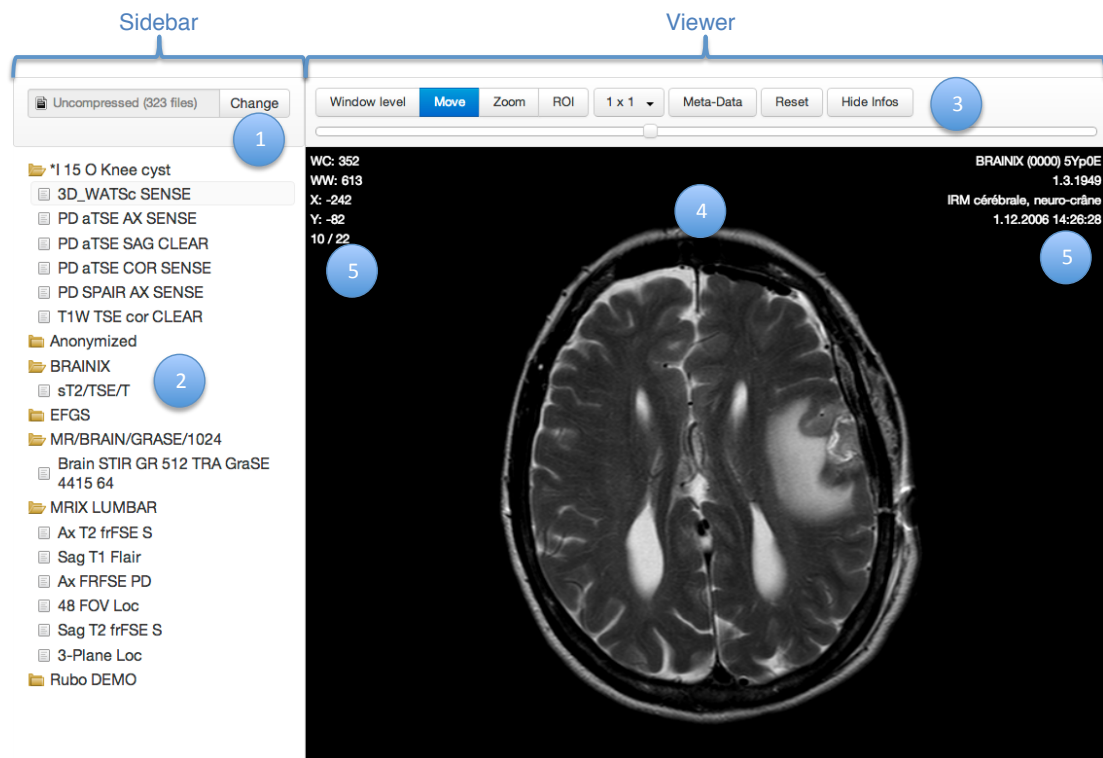


Figure 4.1: User interface of the DICOM Web Viewer consists of a Sidebar with a File Input (1) and a Tree-view (2), and the Viewer with a Widget-Header (3), the actual image (4) and important information displayed in the corners (5).

the patient's name. Parent items are depicted with a folder icon and child items with a file icon (number 2 in Figure 4.1). Also eventual error messages are displayed as red boxes in the sidebar. The file input shows the name of the selected directory and the number of files in it (number 1 in Figure 4.1).

By selecting a study in the sidebar, all images of the study are displayed in the right part of the viewer. It is possible to scroll through all images by using the mouse wheel or the slider in the header (number 3 in Figure 4.1). The DICOM image is displayed according to the size of the browser window, so that always the whole image is shown.

Important information is displayed in the corners of the image: the left corner shows the current Window Center (WC) and Window Width (WW), the actual position (X, Y) of the mouse pointer and the position of the current image in the series. The right corner shows the patient's name, sex and patient's ID in the first row. The second lists patient's birthdate, the third row shows the description of the series and the last one date and time of the series (number 5 in Figure 4.1). This information can be hidden by clicking the "Hide Infos"-button in the header.

In the following the header-widgets (number 3 in Figure 4.1) and their functionalities are described:

- **Window Level:** By selecting this functionality, the parameters of the windowing function - window center and window width - can be altered. Therefore the user moves the mouse pointer to the image, presses and holds down the left button of the mouse, moves the pointer along the x-axis or y-axis and releases the button at the end. "Dragging" along the x-axis changes the value of the window center, along the y-axis the value of the window width. A shortcut for the functionality is provided by *Alt* + left click. Figure 4.2 illustrates the effect of changes in window center- and window width-values on the same image.
- **Move:** This functionality lets the user move the image on the viewer screen by drag-and-drop. The user starts with a click on the image and holds down the left mouse button. The

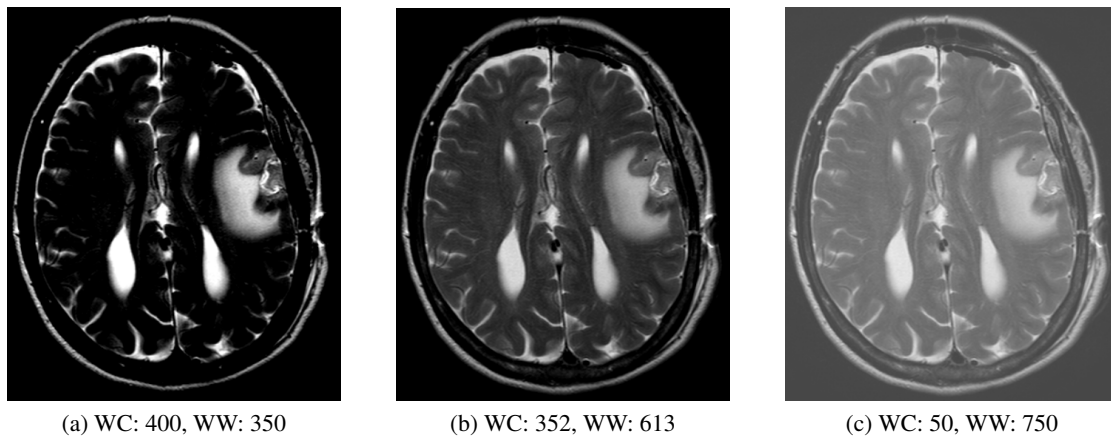


Figure 4.2: Images with different values for Window Center (WC) and Window Width (WW).

image can now be panned as long as the user presses the mouse button. With the release of it the image gets fixed on the actual position. This functionality can be activated by the *cmd-* or *Strg-*shortcut.

- **Zoom:** When this functionality is activated, the user can zoom in or out of the image by drag-and-drop. Dragging upwards enlarges the image whereas dragging downwards shrinks it. Shortcut for this widget is the *Shift*-key.
- **ROI:** This button activates the length measurement functionality. The user selects a start point by clicking on the image and moves the mouse with the left button pressed to an end point. A green line gets drawn on the image during the user interaction to facilitate a more precise measurement. As the mouse button is released the distance between start and end point is displayed on the image. Figure 4.3 illustrates this functionality. The *Caps Lock*-key activates this functionality.
- **1×1:** By clicking on this button a drop-down menu appears and lets the user choose between different grid-layouts. 1×1, 1×2, 2×2 or 4×4 view can be selected. Up to 16 images can then be displayed simultaneously and side by side. The images are shown in the order as they have in the DICOM series. Figure 4.4 illustrates the 2×2 grid-view.
- **Reset:** By clicking on this button, window center and window width are set back to their initial values and the image is drawn renewed in standard size and in the center of the viewer.

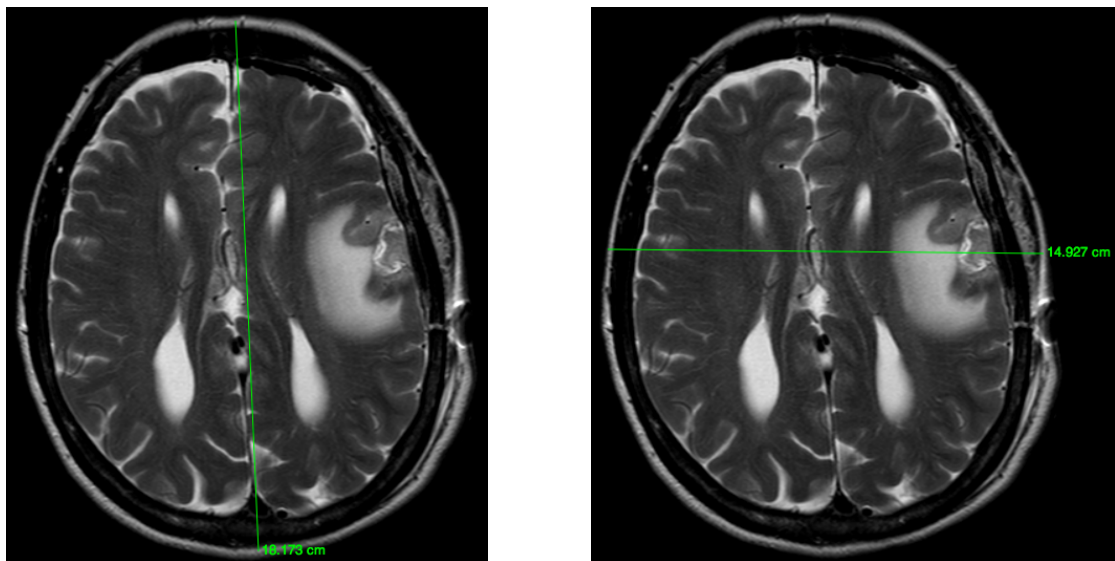


Figure 4.3: The length measurement functionality uses the DICOM “PixelSpacing” attribute to compute the distance between to selected points.

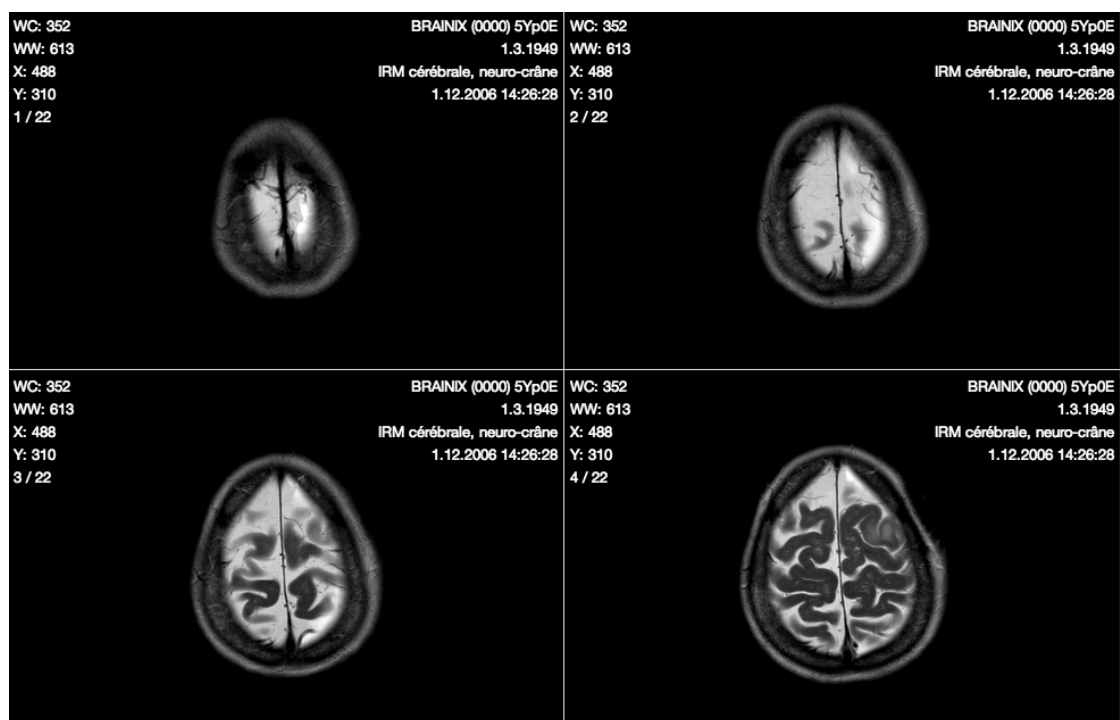


Figure 4.4: The viewer supports 1×1 , 1×2 , 2×2 and 4×4 grid-view. In this case the 2×2 grid-view is chosen and 4 consecutive images of a series are displayed in a row.

- **Hide Infos:** Normally important information is displayed in the corners of the image. By clicking on this button, the infos are hidden and not shown any more. This can be undone by clicking the button again.
- **Slider:** The slider can be used to scroll through all images of a series. For this purpose the user can use the mouse wheel or the slider in the header (number 3 in Figure 4.1).
- **Meta-Data:** This button opens a dialog that lists all DICOM attributes of the current image. The first column shows the names of the attributes and the second column its actual values. Figure 4.5 depicts such a dialog with DICOM attributes.

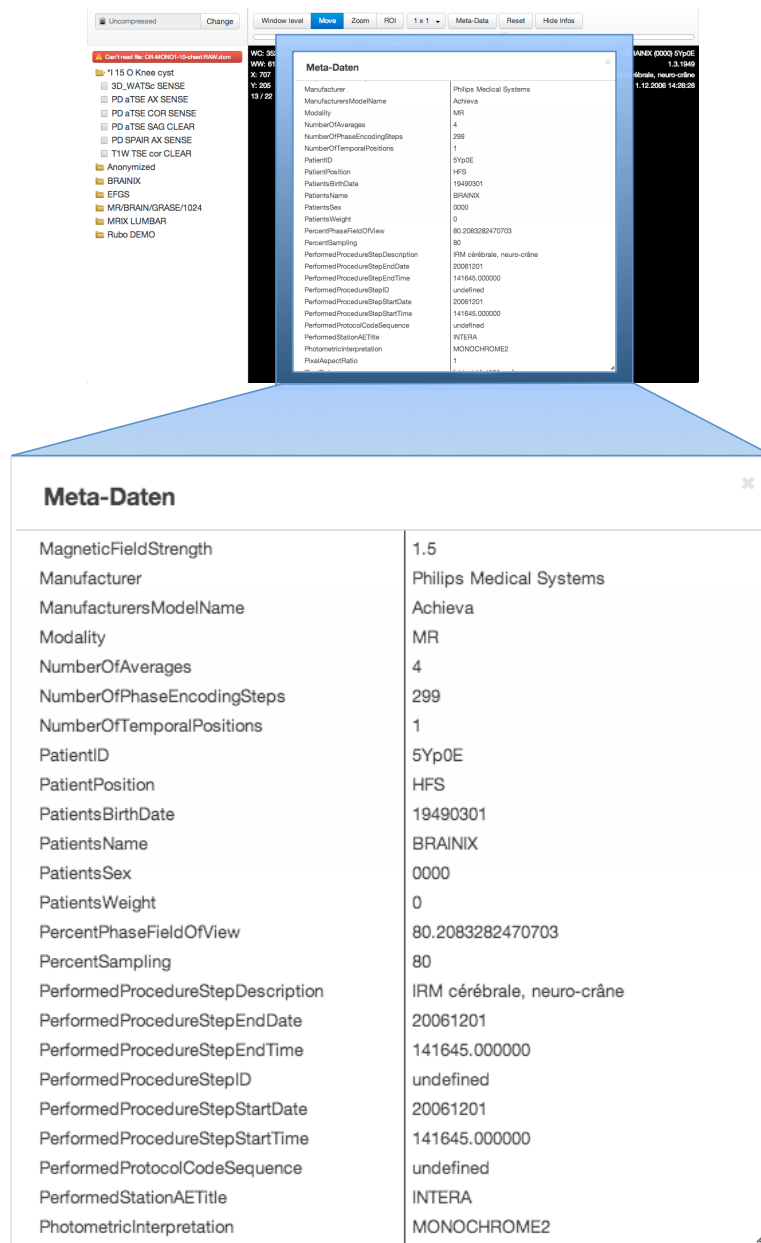


Figure 4.5: When the user clicks the “Meta-Data” button, a dialog with all DICOM attributes of the actual image pops up. It shows all names and values of the attributes, e. g. PatientsBirthDate, PatientsName or Modality.

Conclusion and Future Work

5.1 Conclusion

In the present thesis, first an overview of the DICOM standard and its key concepts has been given. The history of the standard has been outlined, followed by the description of its data format and encoding. Next, the DICOM Service Class has been presented in order to specify the communication between medical devices based on the ISO/OSI-model.

In addition to that, the relation to the HL7 standard and the IHE initiative has been described. Furthermore, the medical imaging workflow has been presented with the intention to demonstrate how these technologies interact in the clinical use.

The last part of the thesis is devoted to implementation aspects of our viewer. The requirements specification, the architecture description and the technology stack has been presented. Furthermore, a detailed overview of our viewer's user interface has been given. All functionalities has been described using several screenshots. In addition a description of the source code and third-party libraries has been annexed.

5.2 Future Work

A web-based service for accessing and retrieving medical images from a PACS server should be investigated. Part 18 of the DICOM standard, "Web Access to DICOM Persistent Objects (WADO)", specifies a service enabling a web client to retrieve DICOM Persistent Objects managed by a web-enabled DICOM server through the HTTP/HTTPS protocol. Data could be retrieved either in a presentation-ready form, e.g. as JPEG image, or in a native DICOM format. The WADO connection to the server could be implemented with the new JavaScript Web Worker API¹.

Future aspects related to lossless compression and decompression of DICOM images should be taken into account. Our viewer supports only uncompressed DICOM files. Libraries for

¹<http://www.whatwg.org/specs/web-apps/current-work/multipage/workers.html>

compression standards like JPEG-LS and JPEG2000 should be integrated in order to support a widespread range of possible DICOM formats.

Source Code

This section gives an overview of the DICOM Web Viewer project and its source code. In the following only the rough file structure, third party libraries and the most important files will be described in detail. The complete source code is available on GitHub (see <http://github.com/mi-kas/webDICOM>). A demo of the DICOM Web Viewer can also be found on <http://mi-kas.github.io/webDICOM/demo/>.

A.1 File Structure

```
DICOM Web Viewer
├── css ..... Folder with CSS-files
├── img ..... Folder with image files
├── index.html
├── js
│   ├── CanvasPainter.js
│   ├── DcmViewer.js
│   ├── FileParser.js
│   ├── Progress.js
│   ├── Toolbox.js
│   ├── Tree.js
│   ├── tools
│   │   ├── Move.js
│   │   ├── Roi.js
│   │   ├── WindowLevel.js
│   │   └── Zoom.js
└── libs ..... Directory with all third-party libraries
```

index.html

This is the default HTML-file that appears in a browser when a user invokes the DICOM Web Viewer. All JavaScript- and CSS-files are loaded within this file. When the Document Object Model (DOM) is fully loaded, event listeners are added to the user interface elements such as buttons, slider or drop-down menus. The index.html file contains also following HTML code:

```
<input id="dicomUpload" type="file" webkitdirectory multiple />
```

This is the the file input, where the user can select the DICOM directory. It makes use of the new HTML5 File API. Not all browsers support the HTML5 tag *webkitdirectory* yet. The *multiple* tag is a fallback for browsers, which don't support the directory upload. At the time of writing only Chrome and Opera Next supported the *webkitdirectory* tag.

DcmViewer.js

This file is the controller of the viewer. It handles all user events and calls the respective functions of the tools.

FileParser.js

This JavaScript file implements the HTML5 File API and makes use of the its FileReader interface. Only DICOM files are read and then parsed by the jsdicom-lib.

CanvasPainter.js

This script takes the parsed DICOM file, applies the modality transformation & windowing function (see Algorithm A.1) and draws the pixel values on the HTML <canvas> element of the viewer. Zoom and pan of the image can also be set.

Progress.js

Progress.js is a JavaScript web-worker that runs in the background, independently of other scripts, without affecting the performance of the page and is used to animate the progress bar during the parsing step.

Tree.js

This file takes as input a list of parsed DICOM files and groups them by patient's name and study's name in a JavaScript object tree that gets rendered in a HTML unordered list. The jqueryTree library displays this list in a graphical tree structure.

Toolbox.js

The toolbox script forwards the user events to the activated tool. For example if the "Move" widget is activated, this tool gets all events such as click, mousemove, mousedown, mouseup, mouseout or mousewheel.

Input : Array A with pixel values, $windowCenter$, $windowWidth$, $rescaleSlope$, $rescaleIntercept$

Output: Array with transformed pixel values

```
1 lowestValue  $\leftarrow windowCenter - windowWidth / 2$ ;  
2 highestValue  $\leftarrow windowCenter + windowWidth / 2$ ;  
3 for  $i = 0$  to  $A.length$  do  
    // modality transformation  
4 intensity  $\leftarrow A[i] \times rescaleSlope + rescaleIntercept$ ;  
    // windowing function  
5 intensity  $\leftarrow (intensity - lowestValue) / (highestValue - lowestValue)$ ;  
6 if intensity  $> 1$  then  
7     intensity  $\leftarrow 1$ ;  
8 else if intensity  $< 0$  then  
9     intensity  $\leftarrow 0$ ;  
10 end  
11  $A[i] \leftarrow intensity \times 255$ ;  
12 end
```

Algorithm A.1: Pseudocode of the modality transformation and windowing function

Tools

All tool scripts - Move.js, Roi.js, WindowLevel.js and Zoom.js - must implement specific functions for the user events which are then called from the Toolbox.js script.

A.2 License

Copyright © 2013 Michael Kaserer

The DICOM Web Viewer is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

A.3 Third-party Libraries

Our viewer uses several third-party libraries. All of them are Open Source and make use either of the GNU General Public License or the MIT license. Table A.1 lists all libraries, its version and references to their projects.

Name	Version	Url
jQuery	2.0.2	http://jquery.com/
jQuery UI	1.10.3	http://jqueryui.com/
SelectBoxIt	3.3.0	http://gregfranko.com/jquery.selectBoxIt.js/
jsdicom-lib	69702a0	https://github.com/Infogosoftware/jsdicom-lib
jqueryTree	r36	http://bit.ly/130I0Fd
jQuery-UI-FileInput	87dfca9	http://github.com/filamentgroup/jQuery-Custom-File-Input
jQuery Bootstrap	0.5	http://addyosmani.github.com/jquery-ui-bootstrap

Table A.1: Third-party libraries used in our viewer.

Bibliography

- [1] Agfa–Gevaert. Scheduled Study Workflow: 2. Acquisition – WEB1000 Documentation. <http://bit.ly/ZOjI4U>. Accessed: 2013–05–28.
- [2] W. Dean Bidgood, Steven C. Horii, Fred W. Prior, and Donald E. Van Syckle. Understanding and Using DICOM, the Data Interchange Standard for Biomedical Imaging. *Journal of the American Medical Informatics Association*, 4(3):199–212, 1997.
- [3] David A. Clunie. *DICOM Structured Reporting*. PixelMed Publishing, Bangor, Pennsylvania, 2000.
- [4] Piero Fraternali, Gustavo Rossi, and Fernando Sánchez-Figueroa. Rich Internet Applications. *Internet Computing, IEEE*, 14(3):9–12, 2010.
- [5] H. K. Huang. *PACS and Imaging Informatics: Basic Principles and Applications*. John Wiley & Sons, Hoboken, New Jersey, 2. edition, 2010.
- [6] IHE International. <http://www.ihe.net>. Accessed: 2013–05–28.
- [7] Integrating the Healthcare Enterprise. Scheduled Workflow – IHE Wiki. http://wiki.ihe.net/index.php?title=Scheduled_Workflow. Accessed: 2013–05–28.
- [8] Peter Mildenerger, Marco Eichelberg, and Eric Martin. Introduction to the DICOM standard. *European Radiology*, 12(4):920–927, 2002.
- [9] Peter Mildenerger, Berthold Wein, Hans-Peter Bursig, and Marco Eichelberg. Aktuelle Entwicklungen von DICOM und IHE. *Der Radiologe*, 45(8):682–689, 2005.
- [10] NEMA PS3 / ISO 12052. *Digital Imaging and Communications in Medicine (DICOM) Standard: Part 1: Introduction and Overview*. National Electrical Manufacturers Association, Rosslyn, VA, USA. http://medical.nema.org/Dicom/2011/11_01pu.pdf. Accessed: 2013–05–14.
- [11] NEMA PS3 / ISO 12052. *Digital Imaging and Communications in Medicine (DICOM) Standard: Part 3: Information Object Definitions*. National Electrical Manufacturers Association, Rosslyn, VA, USA. http://medical.nema.org/Dicom/2011/11_03pu.pdf. Accessed: 2013–05–16.

- [12] Oleg S. Pinykh. *Digital Imaging and Communications in Medicine (DICOM): A Practical Introduction and Survival Guide*. Springer-Verlag, Berlin-Heidelberg, 2012.
- [13] W3C. HTML5: A vocabulary and associated APIs for HTML and XHTML. Technical report, W3C, 2012. <http://www.w3.org/TR/html5/>. Accessed: 2013-05-31.
- [14] Wikipedia. Cascading Style Sheets – Wikipedia, The Free Encyclopedia. <https://en.wikipedia.org/wiki/CSS>. Accessed: 2013-05-31.
- [15] Wikipedia. DICOM – Wikipedia, The Free Encyclopedia. <http://en.wikipedia.org/wiki/DICOM>. Accessed: 2013-05-16.
- [16] Wikipedia. Digital Imaging and Communications in Medicine – Wikipedia, The Free Encyclopedia. http://de.wikipedia.org/wiki/Digital_Imaging_and_Communications_in_Medicine. Accessed: 2013-05-17.
- [17] Wikipedia. Health Level 7 – Wikipedia, The Free Encyclopedia. <http://en.wikipedia.org/wiki/HL7>. Accessed: 2013-05-28.
- [18] Wikipedia. JavaScript – Wikipedia, The Free Encyclopedia. <http://en.wikipedia.org/wiki/JavaScript>. Accessed: 2013-05-31.